

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 681.518:004.652

В.П. Часовских

Уральский государственный лесотехнический университет, Институт экономики
и управления, г. Екатеринбург

САЙТ ПРЕПОДАВАТЕЛЯ ВУЗА - РЕАЛЬНОЕ ПРИЛОЖЕНИЕ



Вступивший в действие с 2013 года Федеральный закон «Об образовании в Российской Федерации» определяет новое содержание высшего профессионального образования, новые модели и технологии образовательного процесса в университете. Очевидно, что выполнение этих требований невозможно без применения информационно-коммуникационных технологий в образовательной деятельности и учебном процессе для всех видов и на всех уровнях образования.

Во всем многообразии применения информационно-коммуникационных технологий в высшем профессиональном образовании наименее исследованным и практически не реализованным является уровень преподавателя. Общая структура современного сайта вуза, место сайта преподавателя, выбор среды разработки и эксплуатации нами была рассмотрена ранее (Часовских, Стаин, 2013, 2014а,б).

Современному пониманию сайта преподавателя вуза как важнейшей составляющей в реализации инновационных проектов по разработке и созданию новых моделей, развитию и совершенствованию существующих моделей, механизмов, образовательных программ, технических средств и технологий в области высшего профессионального образования мы посвятим несколько статей. Я не буду приводить аргументы и обосновывать необходимость наличия у каждого преподавателя сайта для ведения образовательной деятельности. Очевидно, что великое многообразие применения возможностей информационно-коммуникационных технологий в образовательном процессе вуза не сможет обойтись без Интернета, а как следствие, и без сайта в той или иной форме. Важным является выбор технологии (относительно простой и не требующей профессиональной квалификации программиста) создания такого сайта и применения в образовательном процессе преподавателя как элемента общей, контролируемой и управляемой образовательной среды вуза.

В нашей стране, да и в мире, наиболее распространенной платформой для приложений является Microsoft. Современной средой для разработки сайтов является ASP.NET MVC. Рассматриваемые примеры созданы в операционной системе Windows 8.1 и ASP.NET MVC 5. Преимущества и возможности ASP.NET MVC 5 рассмотрены в литературе достаточно полно (Макки, 2010; Столбовский, 2011; Мак-Дональд и др., 2011; Фримен, Сандерсон, 2012; Эспозито, 2012, 2013; Зиборов, 2013; Фримен, 2015а,б). Единственным средством разработки сайтов с применением ASP.NET MVC 5

является программный продукт Microsoft Visual Studio 2013. Он содержит все необходимое для разработки, отладки (включая сервер и базу данных SQL) и публикации сайта. В нашем университете доступна версия Visual Studio Premium 2013, но можно использовать бесплатную версию Visual Studio Express 2013 для Web, которую можно скачать с серверов Microsoft как русифицированную (http://www.visualstudio.com/ru-ru/downloads/download-visual-studio-vs#DownloadFamilies_2) или нерусифицированную версию <http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-products>.

В терминах Visual Studio создаваемый сайт называется решением. Каждое решение содержит один или несколько проектов. Будем рассматривать вариант решения с двумя проектами.

Создание в Visual Studio решения и двух проектов сайта преподавателя

Запускаем Visual Studio и создаем решение SaitPrepod как показано на рис. 1.

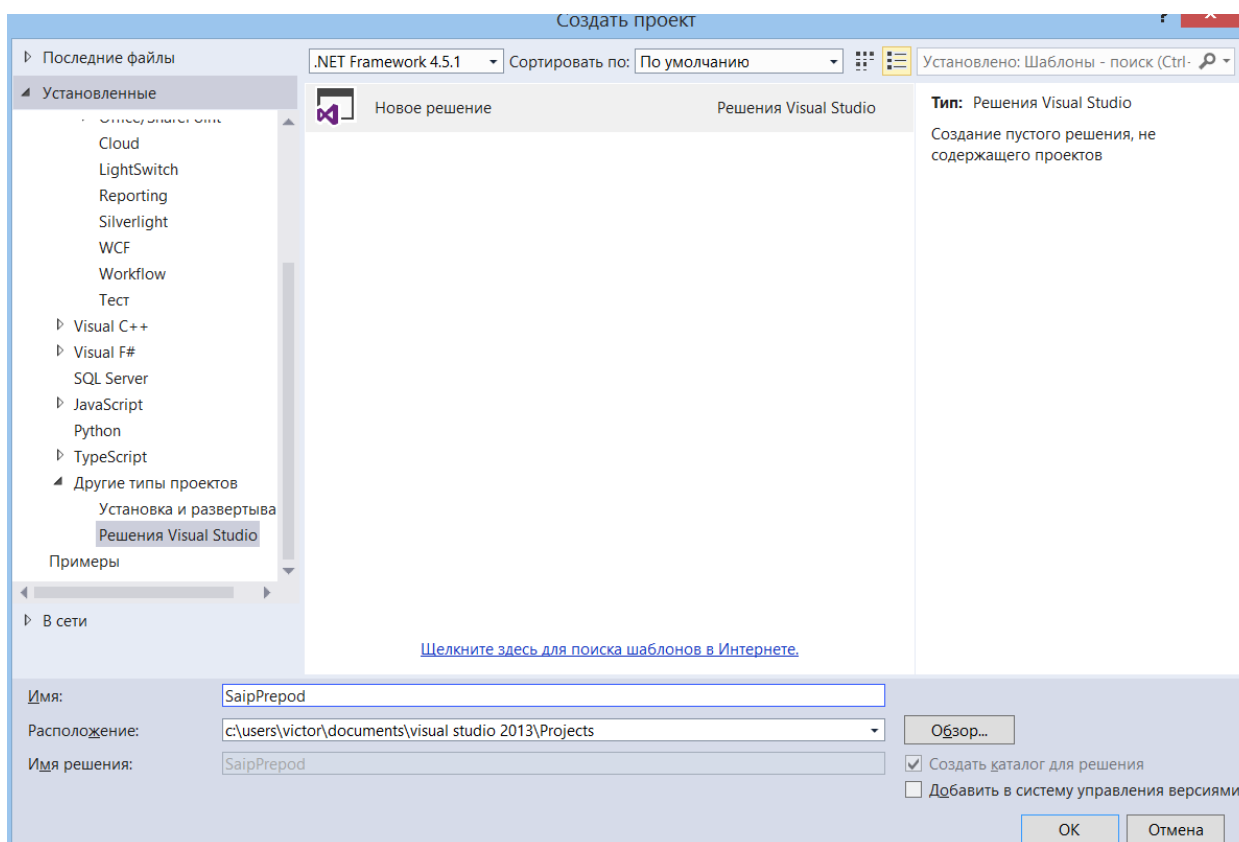


Рис. 1. Создание решения в Visual Studio с именем SaitPrepod.

После нажатия кнопки ОК появится решение SaitPrepod как показано на рис. 2.

С точки зрения структуры сайта решение является контейнером для наших будущих проектов. Первый проект - SaitPrepod.Domain будет предназначен для описания предметной области преподавателя с помощью среды Entity Framework. Второй проект - SaitPrepod.SaitUI будет содержать все необходимое (контроллеры, представления, папки с данными и дополнительными программами, созданными вне среды Visual Studio для публикации сайта и его доступности в Интернет.

Для создания проекта SaitPrepod.Domain необходимо щелкнуть правой кнопкой мыши на тексте – Решение «SaitPrepod» в обозревателе «Решение» (см. рис. 2). Появятся формы Visual Studio для создания проекта (рис. 3).

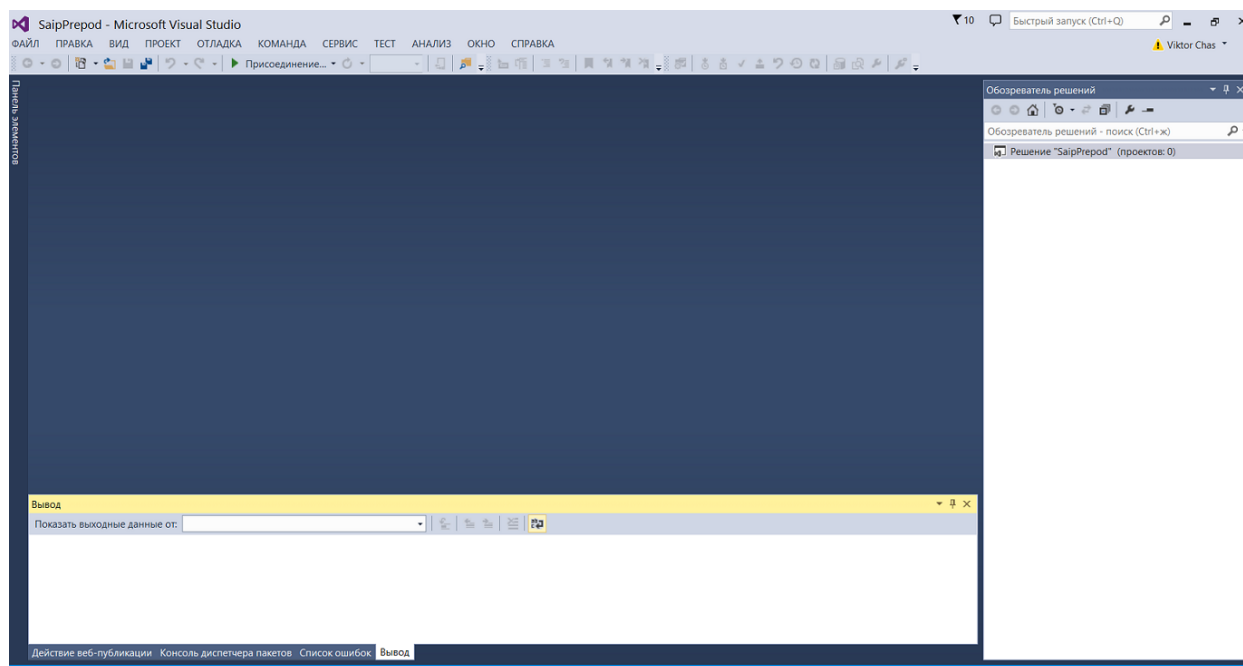


Рис. 2. Решение SaipPrepod.

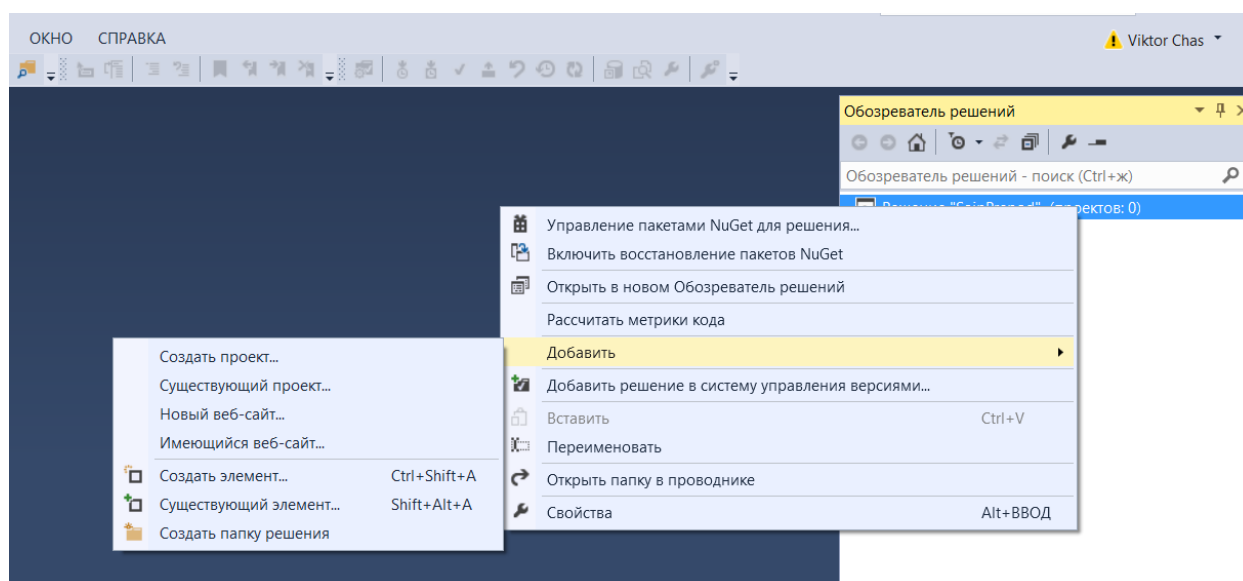


Рис. 3. Указание на создание проекта

Необходимо щелкнуть на тексте «Создать проект». Появится форма Visual Studio, показанная на рис. 4.

Необходимо выбрать язык программирования C# и библиотеку классов, после чего щелкнуть кнопку ОК. Будет создан проект (рис. 5). Файл Class.cs нам не понадобится, его можно удалить. Для этого на имени файла необходимо щелкнуть правой кнопкой мыши и выбрать действие «удалить».

Создание проекта SaipPrepod.SaitUI выполняется аналогично, выбирается язык программирования C# и проект Веб-приложение ASP.NET, как показано на (рис. 6). После нажатия кнопки ОК появляется следующая форма Visual Studio (рис. 7). Необходимо указать MVC, т.е. оставить все без изменения и нажать кнопку ОК. Появится решение с двумя проектами (рис. 8).

Файл «Ваше приложение ASP.NET» не нужен, и его следует удалить. Для этого необходимо щелкнуть правой кнопкой мыши по названию Project_Readme.html и выполнить команду «удаление». Проект SaitPrepod.SaitUI будет в дальнейшем опубликован как сайт, поэтому он должен быть объявлен запускаемым проектом. Для этого необходимо щелкнуть по имени проекта правой кнопкой мыши и выбрать «Назначить запускаемым проектом». Окончательный вид должен соответствовать показанному на рис. 9.

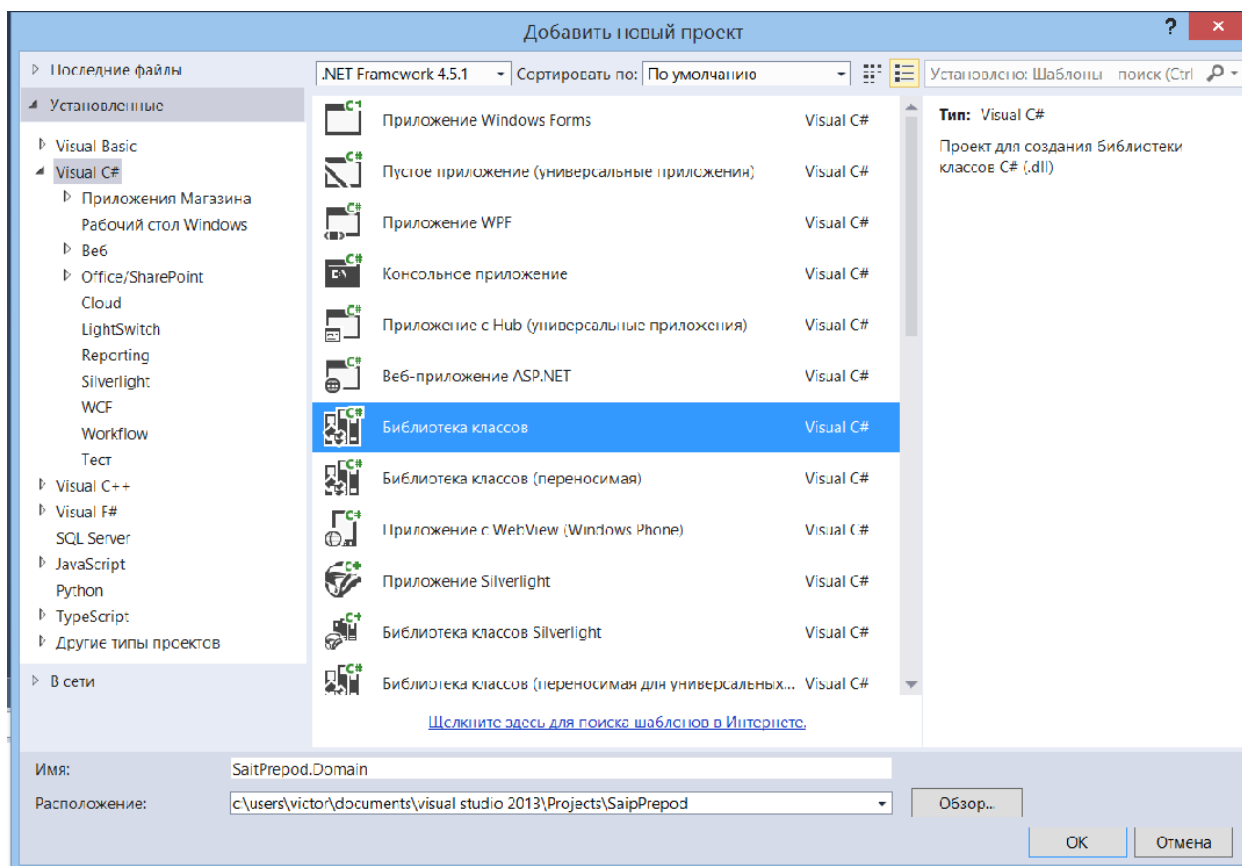


Рис. 4. Создание проекта SaitPrepod.Domain.

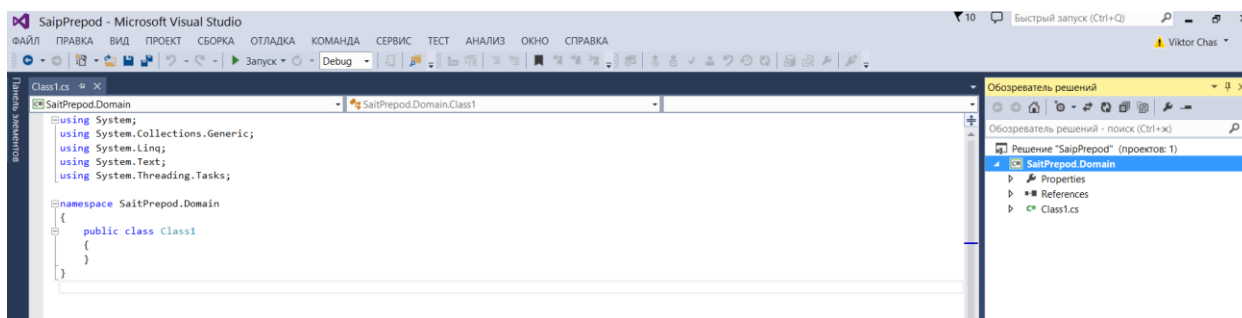


Рис. 5. Проект SaitPrepod.Domain.

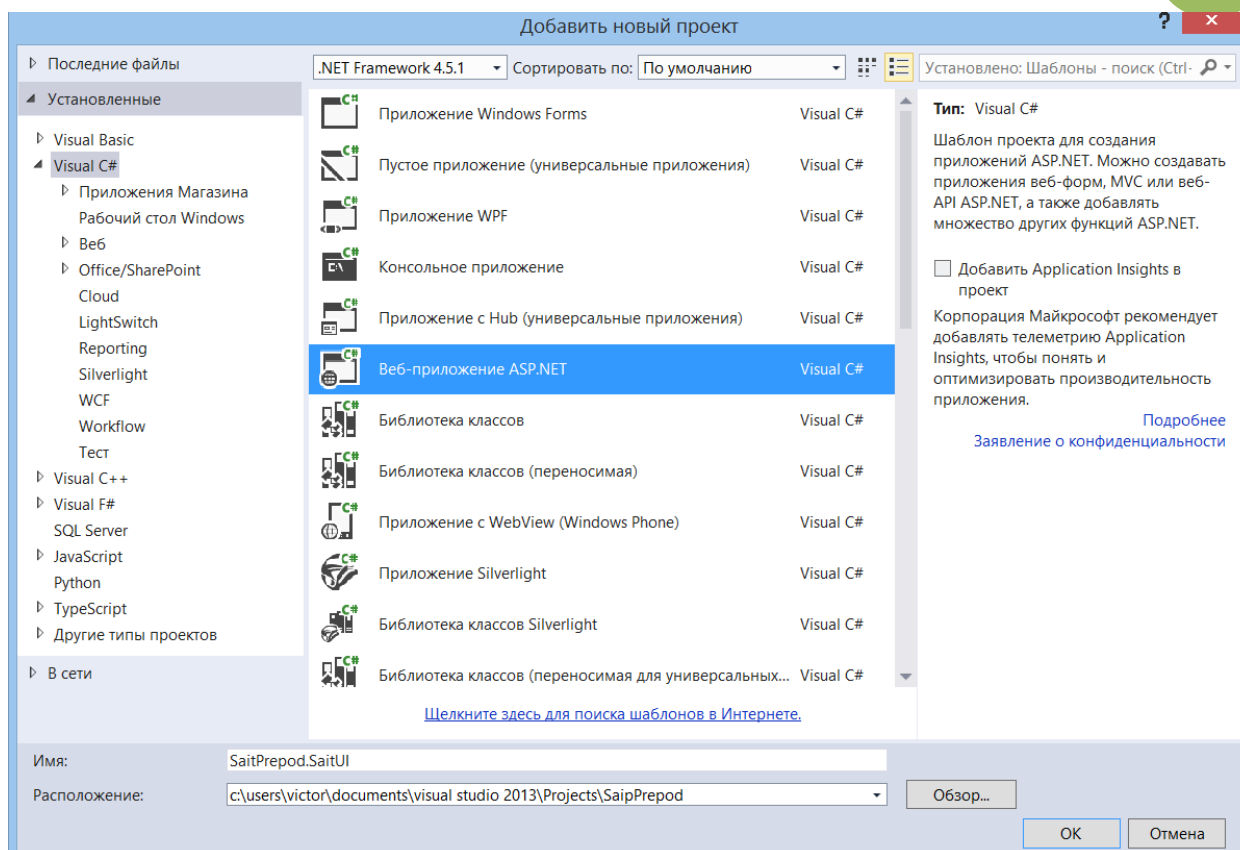


Рис. 6. Свойства проекта SaitPrepod.SaitUI.

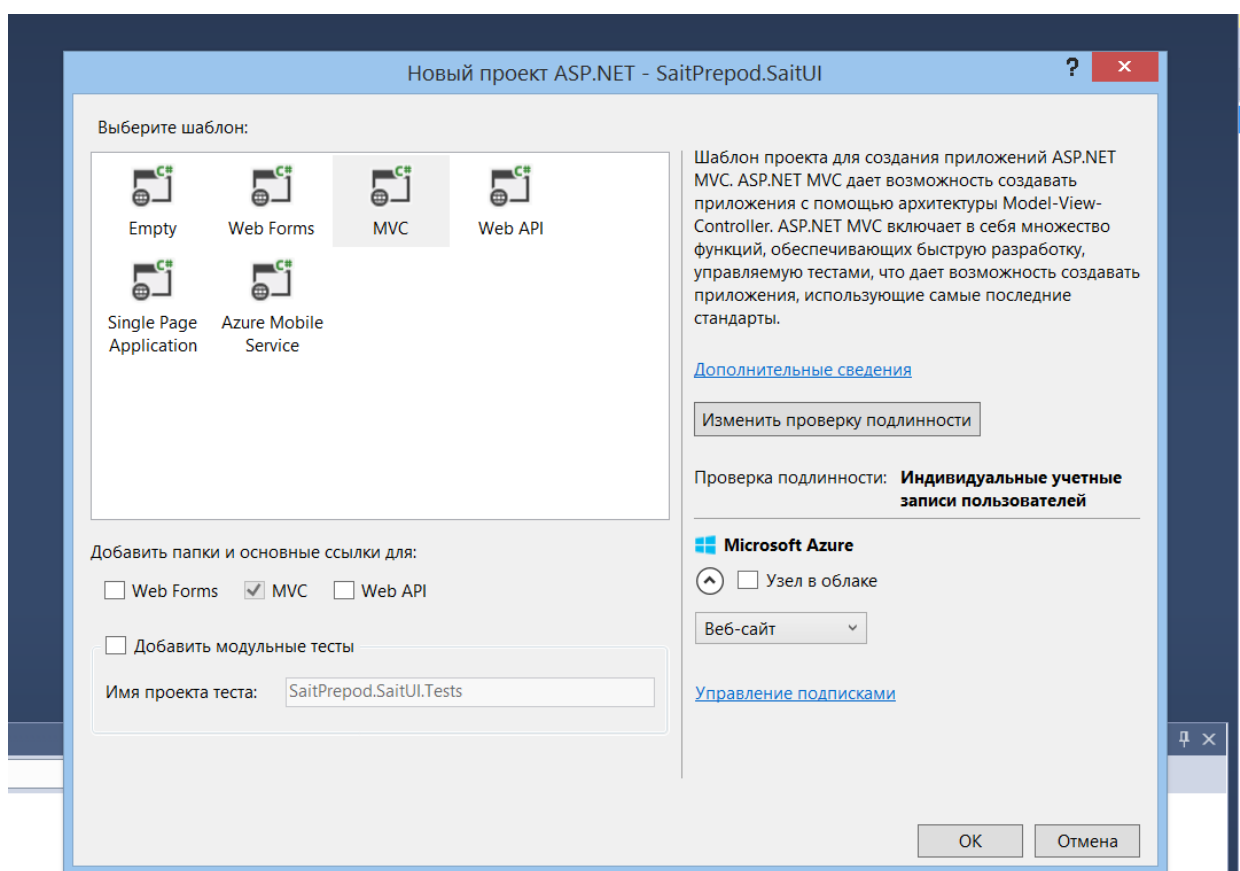


Рис. 7. Выбор формы проекта.

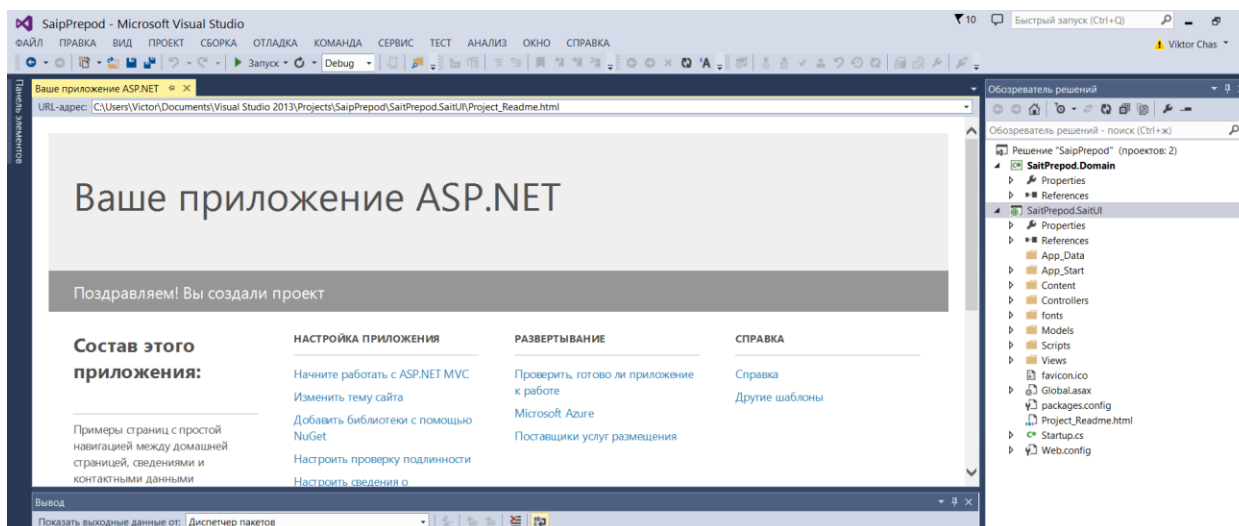


Рис. 8. Решение после создания второго проекта.

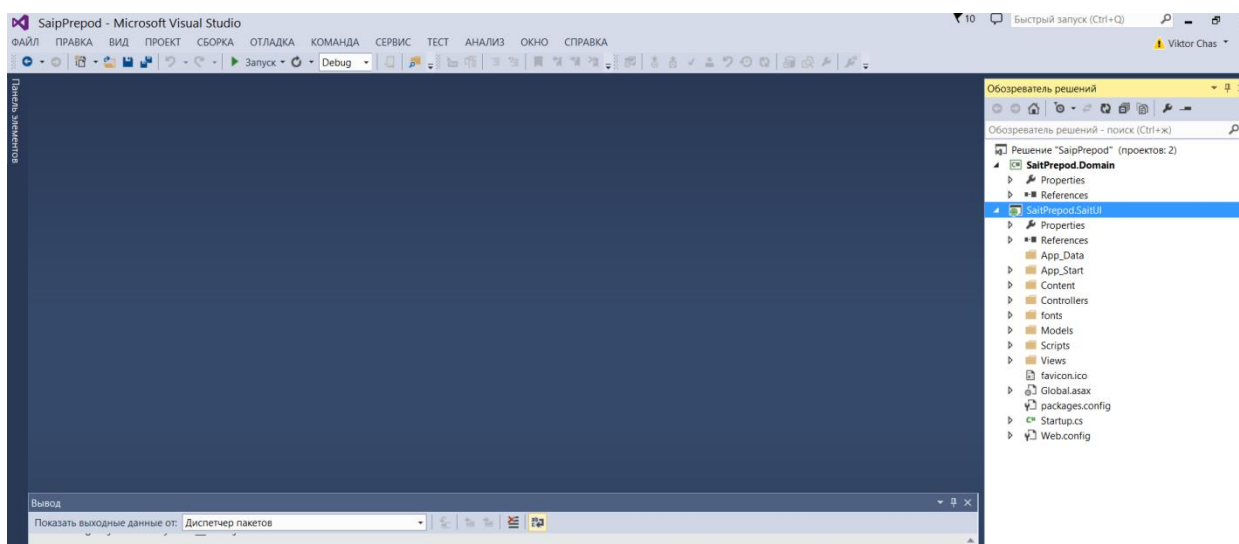


Рис. 9. Решение Saip.Prepod и два проекта SaipPrepod.Domain и SaipPrepod.SaitUI.

Модель и логика предметной области сайта преподавателя

Поскольку мы строим сайт преподавателя вуза, очевидно, что сущностями предметной области является все то, что образует образовательный процесс. К наиболее важным сущностям отнесем следующие: студенты; научные исследования; читаемые дисциплины; контрольные работы; курсовые работы; лабораторно-практические занятия; самостоятельная работа студента; практики; выпускная квалификационная работа; консультации; учебно-методическое обеспечение (обозначим как библиотека); расписание занятий; контактная информация; состояние учебы студента. Не претендуя на функциональную полноту, ограничимся этим перечнем и предусмотрим возможность функционального расширения сайта.

Элементы, образующие перечисленные сущности предметной области, приведем в табличной форме (табл. 1). Связь между сущностями укажем с помощью ее номера.

Таблица 1

Элементы сущностей предметной области

№ пп	Сущность	Элементы сущности	Связь между сущностями
1.	Студент	Фамилия; имя; отчество; форма обучения; уровень образования; форма оплаты; группа; дата зачисления.	17, 18, 23
2.	Дисциплина	Название; зачетные единицы; курсовая работа; контрольная работа; форма получения оценки; рабочая программа.	
3.	Литература дисциплины	Дисциплина; тип литературы; название; автор(ы); издание; год.	3, 22
4.	НИР магистров	Образовательная программа; магистерская программа; научное направление.	
5.	Научные направления для студентов	Первое направление; второе направление; третье направление; четвертое направление; пятое направление.	
6.	НИР выпускающей кафедры	Название НИР; № гос. регистрации; срок выполнения; полученные результаты; исполнители; интеллектуальный продукт.	
7.	Полученный интеллектуальный продукт	Тип документа; название; номер; приоритет; правообладатели; авторы; номер заявки; дата регистрации	
8.	Практика студентов	Студент; название практики; задание на практику; место практики; дата – начало практики; дата – окончания практики; дата – защита отчета; оценка.	1, 19
9.	Публикации	Тип публикации; уровень публикации; название; авторы; издание; год.	21, 24
10.	Расписание	Форма обучения; семестр; расписание.	23
11.	Темы контрольных работ	Тема (задача) контрольной работы; тип работы; студент; дисциплина.	1, 2, 21
12.	Темы курсовых работ	Тема курсовой работы; тип работы; студент; дисциплина.	1, 2, 21
13.	Выбор контрольных работ	Дисциплина; номер работы; форма обучения; уровень образования.	2, 17, 20, 23
14.	Учеба студента	Студент; семестр; дисциплина; курсовая работа – да (нет); оценка за курсовую работу; контрольная работа – да (нет); контрольная работа - оценка; лабораторно-практические занятия да - нет; число занятий; оценка; самостоятельная работа да-нет; тема; оценка; зачет да-нет; оценка; экзамен да-нет; оценка; зачетных единиц.	1, 2, 25

Продолжение таблицы 1

№ пп	Сущность	Элементы сущности	Связь между сущностями
15.	Выпускная квалификационная работа	Студент; тема работы; дата утверждения; дата окончания; дата защиты; оценка; протокол; заключение о плагиате; заключение руководителя; текст работы; дата защиты; оценка за защиту; протокол защиты.	1
16.	Консультация: вопрос - ответ	Фамилия; вопрос; ответ; да - нет модератора.	
17.	Форма обучения	Форма обучения.	
18.	Форма оплаты	Форма оплаты.	
19.	Название практик	Название практик.	
20.	Номер контрольной работы	Номер контрольной работы.	
21.	Тип работы	Тип работы.	
22.	Тип научной литературы	Тип научной литературы.	
23.	Уровень образования	Уровень образования.	
24.	Уровень публикации	Уровень публикации.	
25.	Да-нет	Да-нет.	

Формальное описание сущностей выполним в языке C#, добавим в эти описания атрибуты метаданных (указываются в квадратных скобках).

Сущность «Студент»:

```
public class Student
{
    [DatabaseGenerated(DatabaseGeneratedOption.None)]
    [Display(Name = "Номер студенческого билета")]
    public int StudentID { get; set; }
    [MaxLength(50, ErrorMessage = "В фамилии можно указать до 50 символов.")]
    [Display(Name = "Фамилия")]
    public string Family { get; set; }
    [Display(Name = "Имя")]
    public string Imy { get; set; }
    [Display(Name = "Отчество")]
    public string Otchestvo { get; set; }
    [Display(Name = "Форма обучения")]
    public int _0FormaObucheniyaID { get; set; }
    [Display(Name = "Уровень образования")]
    public int _0UrovenObrazovaniyaID { get; set; }
    [Display(Name = "Форма оплаты")]
    public int _0FormaOplatyID { get; set; }
    [Display(Name = "Группа")]
    public string Gruppya { get; set; }
    [Display(Name = "Дата зачисления")]
    [DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true)]
    public DateTime DataZachisleniy { get; set; }

    [Display(Name = "Фамилия и имя")]
    public string FullName
    {
        get
    }
```



```

    {
        return Family + ", " + Imy;
    }
}

public virtual _0FormaObucheniy _0FormaObucheniy { get; set; }
public virtual _0FormaOplati _0FormaOplati { get; set; }
public virtual _0UrovenObrazovaniy _0UrovenObrazovaniy { get; set; }
public virtual ICollection<Student> Students { get; set; }
}

```

Сущность «Дисциплина»:

```

public class Disziplina
{
    [HiddenInput(DisplayValue = false)]
    [Display(Name = "Ключ записи")]
    public int DisziplinaID { get; set; }
    [Display(Name = "Название")]
    public string Nazvanie { get; set; }
    [Display(Name = "Зачетные единицы")]
    public string ZachetEdin { get; set; }
    [Display(Name = "Курсовая работа")]
    public string KursRabota { get; set; }
    [Display(Name = "Контрольная работа")]
    public string KontRabota { get; set; }
    [Display(Name = "Форма получения оценки")]
    public string FormaOzenki { get; set; }
    [Display(Name = "Рабочая программа")]
    public string ObrzavProgramma { get; set; }
    [Display(Name = "Имя файла рабочей программы")]
    public string DDopPole01 { get; set; }
    [Display(Name = "Доп. поле 02")]
    public string DDopPole02 { get; set; }
    public virtual ICollection<Disziplina> Disziplinas { get; set; }
}

```

Сущность «Литература дисциплины»:

```

public class LiterDisz
{
    public int LiterDiszID { get; set; }
    [Display(Name = "Дисциплина")]
    public int DisziplinaID { get; set; }
    [Display(Name = "Тип литературы")]
    public int _0TipUchNauchLiteraturiID { get; set; }
    [Display(Name = "Название")]
    public string Nazvanie { get; set; }
    [Display(Name = "Авторы")]
    public string Avtor { get; set; }
    [Display(Name = "Издание")]
    public string Izdanie { get; set; }
    [Display(Name = "Год")]
    public string God { get; set; }
    [Display(Name = "Файл загрузки")]
    public string Zagruzit { get; set; }
    public virtual Disziplina Disziplina { get; set; }
    public virtual _0TipUchNauchLiteraturi _0TipUchNauchLiteraturi { get; set; }
    public virtual ICollection<LiterDisz> LiterDiszs { get; set; }
}

```

Сущность «НИР магистров»:

```
public int NauchIssledMagistrovID { get; set; }

[DisplayName("Образовательная программа магистра")]
public string OPMagistra { get; set; }
[DisplayName("Магистрская (профиль) программа")]
public string MagistrProgramma { get; set; }
[DisplayName("Научное направление")]
public string NauchNapravlenie { get; set; }
[Display(Name = " Ссылка первого столбца")]
public string Dop01NauchIssledMagistrov { get; set; }
[Display(Name = " Ссылка второго столбца")]
public string Dop02NauchIssledMagistrov { get; set; }
[Display(Name = " Ссылка третьего столбца")]
public string Dop03NauchIssledMagistrov { get; set; }

}
```

Сущность «Научные направления для студентов»:

```
public class NauchIssledov
{
    public int NauchIssledovID { get; set; }

    [DisplayName("Первое научное направление")]
    public string NauchInteres01 { get; set; }
    [DisplayName("Второе научное направление, если отсутствует, то Нет")]
    public string NauchInteres02 { get; set; }
    [DisplayName("Третье научное направление, если отсутствует, то Нет")]
    public string NauchInteres03 { get; set; }
    [DisplayName("Четвертое научное направление, если отсутствует, то Нет")]
    public string NauchInteres04 { get; set; }
    [DisplayName("Пятое научное направление, если отсутствует, то Нет")]
    public string NauchInteres05 { get; set; }
    [Display(Name = "Научные направления, разрабатываемые автором")]
    public string Dop01NauchInteres { get; set; }
    [Display(Name = "Dop02NauchInteres")]
    public string Dop02NauchInteres { get; set; }
    [Display(Name = "Dop03NauchInteres")]
    public string Dop03NauchInteres { get; set; }
}
```

Сущность «НИР выпускающей кафедры»:

```
public class Nir
{
    public int NirID { get; set; }

    [DisplayName("Название НИР")]
    public string NazvanieNir { get; set; }
    [DisplayName("№ гос. регистрации")]
    public string GosReg { get; set; }
    [DisplayName("Срок выполнения")]
    public string BeginEnd { get; set; }
    [DisplayName("Полученные результаты")]
    public string Rezultati { get; set; }
    [DisplayName("Исполнители")]
    public string Ispolzovanie { get; set; }
    [DisplayName("Интеллектуальный продукт 1")]
    public string IntelProduct { get; set; }
}
```

}

Сущность «Полученный интеллектуальный продукт»:

```
public class IntelProduct
{
    public int IntelProductID { get; set; }
    [Display(Name = "Тип документа")]
    public string TipDocumenta { get; set; }
    [Display(Name = "Название")]
    public string Nazvanie { get; set; }
    [Display(Name = "Номер")]
    public string Nomer { get; set; }
    [Display(Name = "Приоритет")]
    public string Pijritet { get; set; }
    [Display(Name = "Правообладатели")]
    public string Pravooblad { get; set; }
    [Display(Name = "Авторы")]
    public string Avtori { get; set; }
    [Display(Name = "Номер заявки")]
    public string NomerZay { get; set; }
    [Display(Name = "Дата регистрации")]
    public string DataReg { get; set; }
    [Display(Name = "Файл загрузки")]
    public string Zagruzit { get; set; }
}
```

Сущность «Практика студентов»:

```
public class Praktika
{
    public int PraktikaID { get; set; }
    [DisplayName("Студент")]
    public int StudentID { get; set; }
    [DisplayName("Название практики")]
    public int _0NazvaniePraktikID { get; set; }
    [DisplayName("Задание на практику")]
    public string ZadaniePraktiki { get; set; }
    [DisplayName("Место практики")]
    public string MestoPraktiki { get; set; }
    [DisplayName("Дата - начало практики")]
    [DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true)]
    public DateTime NachaloPraktiki { get; set; }
    [DisplayName("Дата - окончание практики")]
    [DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true)]
    public DateTime OkonchaniePraktiki { get; set; }
    [DisplayName("Дата - защита отчета о практике")]
    [DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true)]
    public DateTime ZasitaPraktiki { get; set; }
    [DisplayName("Оценка")]
    public string OzenkaPraktiki { get; set; }
    [DisplayName("Имя файла отчета")]
    public string FailPraktiki { get; set; }

    public virtual Student Student { get; set; }
    public virtual _0NazvaniePraktik _0NazvaniePraktik { get; set; }
    public virtual ICollection<Praktika> Praktikas { get; set; }
}
```

Сущность «Публикации»:

```
public class Publikazii
{
    public int PublikaziiID { get; set; }
    [Display(Name = "Тип литературы")]
    public int _0TipUchNauchLiteraturiID { get; set; }
    [Display(Name = "Уровень публикации")]
    public int _0UrovenPublikaziiID { get; set; }
    [Display(Name = "Название")]
    public string Nazvanie { get; set; }
    [Display(Name = "Авторы")]
    public string Avtor { get; set; }
    [Display(Name = "Издание")]
    public string Izdanie { get; set; }
    [Display(Name = "Год")]
    public string God { get; set; }
    [Display(Name = "Файл загрузки")]
    public string Zagruzit { get; set; }
    public virtual _0TipUchNauchLiteraturi _0TipUchNauchLiteraturi { get; set; }
    public virtual _0UrovenPublikazii _0UrovenPublikazii { get; set; }

    public virtual ICollection<Publikazii> Publikaziis { get; set; }
}
```

Сущность «Расписание»:

```
public class Raspisanie
{
    public int RaspisanieID { get; set; }
    [DisplayName("Форма обучения")]
    public int _0FormaObucheniyID { get; set; }
    [DisplayName("Семестр")]
    public string SemestrRaspisaniy { get; set; }
    [DisplayName("Имя файла")]
    public string ImyFailaRaspisaniy { get; set; }
    [Display(Name = "1 - ая строка заголовка")]
    public string Dop01TRaspisanie { get; set; }
    [Display(Name = "2 - ая строка заголовка")]
    public string Dop02Raspisanie { get; set; }
    public virtual _0FormaObucheniy _0FormaObucheniy { get; set; }
    public virtual ICollection<Raspisanie> Raspisaniis { get; set; }
}
```

Сущность «Темы контрольных работ»:

```
public class TemiKontRab
{
    public int TemiKontRabID { get; set; }
    [DisplayName("Тема(задача) контрольной работы")]
    public string KontRabotNazvanie { get; set; }
    [DisplayName("Тип работы")]
    public int _0TipRabotiID { get; set; }
    [DisplayName("Студент")]
    public int StudentID { get; set; }
    [DisplayName("Дисциплина")]
    public int DisziplinaID { get; set; }
    [Display(Name = "Имя файла")]
    public string KontDopPole01 { get; set; }
    [Display(Name = "Фото или ничего")]
    public string KontDopPole02 { get; set; }
    [Display(Name = "Контрольная работа 3")]
}
```

```
public string KuKontrDopPole05 { get; set; }  
public virtual _0TipRaboti _0TipRaboti { get; set; }  
public virtual Student Student { get; set; }  
public virtual Disziplina Disziplina { get; set; }  
public virtual ICollection<TemiKontRab> TemiKontRabs { get; set; }  
}
```

Сущность «Темы курсовых работ»:

```
public class TemiKursRabot  
{  
    public int TemiKursRabotID { get; set; }  
    [DisplayName("Тема курсовой работы")]  
    public string KursRabotNazvanie { get; set; }  
    [DisplayName("Тип работы")]  
    public int _0TipRabotiID { get; set; }  
    [DisplayName("Студент")]  
    public int StudentID { get; set; }  
    [DisplayName("Дисциплина")]  
    public int DisziplinaID { get; set; }  
    [Display(Name = "Запись с перечнем работ 1")]  
    public virtual _0TipRaboti _0TipRaboti { get; set; }  
    public virtual Disziplina Disziplina { get; set; }  
    public virtual Student Student { get; set; }  
    public virtual ICollection<TemiKursRabot> TemiKursRabots { get; set; }  
}
```

Сущность «Учеба студентов»:

```
public class UchebaStudenta  
{  
    public int UchebaStudentaID { get; set; }  
    [Display(Name = "Студент")]  
    public int StudentID { get; set; }  
    [Display(Name = "Семестр")]  
    public string Semestr { get; set; }  
    [Display(Name = "Дисциплина")]  
    public int DisziplinaID { get; set; }  
    [UIHint("YesNo")]  
    public int _0YesNoID { get; set; }  
    [Display(Name = "Курсовая работа оценка")]  
    public string KursRabOzenka { get; set; }  
    [Display(Name = "Контрольная работа ДА - НЕТ")]  
    [UIHint("YesNo")]  
    public string YesNo1 { get; set; }  
    [Display(Name = "Контрольная работа оценка")]  
    public string KontRabOzenka { get; set; }  
    [Display(Name = "Лабораторно-практические занятия ДА - НЕТ")]  
    [UIHint("YesNo")]  
    public string YesNo2 { get; set; }  
    [Display(Name = "Число лабораторно-практических занятий")]  
    public string ChisloLabPrakt { get; set; }  
    [Display(Name = "Лабораторно-практические занятия оценка")]  
    public string LabPraktOzenka { get; set; }  
    [Display(Name = "Самостоятельная работа ДА - НЕТ")]  
    [UIHint("YesNo")]  
    public string YesNo3 { get; set; }  
    [Display(Name = "Тема самостоятельной работы")]  
    public string TemiSamRab { get; set; }  
    [Display(Name = "Самостоятельная работа оценка")]  
    public string SamRabOzenka { get; set; }  
    [Display(Name = "Зачет ДА - НЕТ")]  
    [UIHint("YesNo")]  
}
```

```

public string YesNo4 { get; set; }
[Display(Name = "Оценка ")]
public string ZachetOzenka { get; set; }
[Display(Name = "Экзамен ДА - НЕТ")]
[UIHint("YesNo")]
public string YesNo5 { get; set; }
[Display(Name = "Оценка ")]
public string EkzamentOzenka { get; set; }
[Display(Name = "Зачетных единиц ")]
public string ZachetEdiniz { get; set; }
public virtual Student Student { get; set; }
public virtual Disziplina Disziplina { get; set; }
public virtual _0YesNo _0YesNo { get; set; }
public virtual ICollection<UchebaStudenta> UchebaStudentas { get; set; }
}

```

Сущность «Выбор контрольной работы»:

```

public class ViborKontRaboti
{
    [HiddenInput(DisplayValue = false)]
    [Display(Name = "Ключ записи")]
    public int ViborKontRabotiID { get; set; }
    [Display(Name = "Дисциплина")]
    public int DisziplinaID { get; set; }
    [Display(Name = "Номер работы")]
    public int _0NomerKontRabotiID { get; set; }
    [Display(Name = "Форма обучения")]
    public int _0FormaObucheniyaID { get; set; }
    [Display(Name = "Уровень образования")]
    public int _0UrovenObrazovaniyaID { get; set; }
    public virtual Disziplina Disziplina { get; set; }
    public virtual _0NomerKontRaboti _0NomerKontRaboti { get; set; }
    public virtual _0FormaObucheniya _0FormaObucheniya { get; set; }
    public virtual _0UrovenObrazovaniya _0UrovenObrazovaniya { get; set; }
    public virtual ICollection<ViborKontRaboti> ViborKontRabotis { get; set; }
}

```

Сущность «Выпускная квалификационная работа»:

```

public class VipuskKvalifRabot
{
    public int VipuskKvalifRabotID { get; set; }
    [Display(Name = "Студент")]
    public int StudentID { get; set; }
    [Display(Name = "Тема ВК работы")]
    public string TemaVKR { get; set; }
    [Display(Name = "Дата - утверждения темы ВКР")]
    [DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true)]
    public DateTime NachaloRazVPR { get; set; }
    [Display(Name = "Дата - окончание разработки ВКР")]
    [DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true)]
    public DateTime OkonchanieRazVKR { get; set; }
    [Display(Name = "Дата - предварительная защита на кафедре")]
    [DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true)]
    public DateTime PredZasitaVKR { get; set; }
    [Display(Name = "Оценка предварительной защиты")]
    public string OzenkaPredZasitaVKR { get; set; }
    [Display(Name = "Протокол предварительной защиты")]
    public string ProtokolPredZasitaVKR { get; set; }
    [Display(Name = "Заключение о плагиате ВКР")]
    public string SpravkaPlagiat { get; set; }
}

```

```
[DisplayName("Заключение руководителя ВКР т.е. моё мнение")]
public string ZaklRukovodVKR { get; set; }
[DisplayName("Выпускная квалификационная работа")]
public string FailVKR { get; set; }
[DisplayName("Дата - защита в ГЭК")]
[DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true)]
public DateTime ZasitaVKRvGEK { get; set; }
[DisplayName("Оценка за защиту в ГЭК")]
public string OzenkaZasitaVKRvGEK { get; set; }
[DisplayName("Протокол защиты в ГЭК")]
public string ProtokolZasitiGEK { get; set; }
public virtual Student Student { get; set; }
public virtual ICollection<VipuskKvalifRabot> VipuskKvalifRabots { get; set; }
}
```

Сущность – «Консультация: вопрос-ответ»:

```
public class VoprosOtvvet
{
    [HiddenInput(DisplayValue = false)]
    [Display(Name = "Ключ записи Вопрос - Ответ")]
    public int VoprosOtvvetID { get; set; }
    [Display(Name = "Номер студенческого билета ")]
    [Required(ErrorMessage = "Пожалуйста укажите номер студенческого билета.")]
    public string NomerStudBileta { get; set; }
    [DataType(DataType.MultilineText)]
    [Required(ErrorMessage = "Пожалуйста сформулируйте свой вопрос.")]
    [Display(Name = "Вопрос")]
    [MaxLength(500, ErrorMessage = "В вопросе можно указать до 500 символов.")]
    public string Vopros { get; set; }
    [Display(Name = "Ответ")]
    [DataType(DataType.MultilineText)]
    public string Otvvet { get; set; }
    [Display(Name = "Дата в формате - дд.мм.гггг ")]
    [DisplayFormat(DataFormatString = "{0:d}", ApplyFormatInEditMode = true
    [Required(ErrorMessage = "Пожалуйста укажите дату в формате дд-мм-гггг.")]
    public DateTime DateAdded { get; set; }
    [Display(Name = "Если вопрос корректен, то указать Да")]
    public string DR1 { get; set; }
}
```

Сущность «Форма обучения»:

```
public class _0FormaObucheni
{
    public int _0FormaObucheniID { get; set; }
    [DisplayName("Формы обучения")]
    public string FormaObucheni { get; set; }
    public virtual ICollection<_0FormaObucheni> _0FormaObucheniys { get; set; }
}
```

Сущность «Форма оплаты»:

```
public class _0FormaOplati
{
    [DisplayName("Ключ")]
    public int _0FormaOplatiID { get; set; }
    [DisplayName("Формы оплаты")]
    public string FormaOplati { get; set; }
    public virtual ICollection<_0FormaOplati> _0FormaOplatis { get; set; }
}
```


Сущность «Название практик»:

```
public class _0NazvaniePraktik
{
    public int _0NazvaniePraktikID { get; set; }
    [DisplayName("Название практики")]
    public string NazvaniePraktik { get; set; }
    public virtual ICollection<_0NazvaniePraktik> _0NazvaniePraktiks { get; set; }
}
```

Сущность «Номер контрольной работы»:

```
public class _0NomerKontRaboti
{
    [DisplayName("Ключ")]
    public int _0NomerKontRabotiID { get; set; }
    [DisplayName("Номер контрольной работы")]
    public string NomerKontRaboti { get; set; }
    public virtual ICollection<_0NomerKontRaboti> _0NomerKontRabotis { get; set; }
}
```

Сущность «Тип работы»:

```
public class _0TipRaboti
{
    [DisplayName("Ключ")]
    public int _0TipRabotiID { get; set; }
    [DisplayName("Тип работы")]
    public string TipRaboti { get; set; }
    public virtual ICollection<_0TipRaboti> _0TipRabotis { get; set; }
}
```

Сущность «Тип литературы»:

```
public class _0TipUchNauchLiteraturi
{
    public int _0TipUchNauchLiteraturiID { get; set; }
    [Display(Name = "Числовое поле")]
    public int TipUchNaucLiteraturi { get; set; }
    [Display(Name = "Тип литературы")]
    public string TipUchNaucLiterPole01 { get; set; }
    [Display(Name = "Доп. поле 02")]
    public string TipUchNaucLiterPole02 { get; set; }
    public virtual ICollection<_0TipUchNauchLiteraturi> _0TipUchNauchLiteraturis {
get; set; }
}
```

Сущность «Уровень образования»:

```
public class _0UrovenObrazovaniy
{
    [DisplayName("Ключ")]
    public int _0UrovenObrazovaniyID { get; set; }
    [DisplayName("Уровень образования")]
    public string UrovenObrazovaniy { get; set; }
    public virtual ICollection<_0UrovenObrazovaniy> _0UrovenObrazovaniys { get;
set; }
}
```

Сущность «Уровень публикации»:

```
public class _0UrovenPublikazii
{
    public int _0UrovenPublikaziiID { get; set; }
    [Display(Name = "Числовое поле")]
    public int UrovenPublikazii { get; set; }
    [Display(Name = "Уровень научной публикации")]
    public string UrovenPublikaziiPole01 { get; set; }
    [Display(Name = "Доп. поле 02")]
    public string UrovenPublikaziiPole02 { get; set; }
    public virtual ICollection<_0UrovenPublikazii> _0UrovenPublikaziis { get; set; }
}
```

Сущность «Да-Нет»:

```
public class _0YesNo
{
    public int _0YesNoID { get; set; }
    [Display(Name = "ДА или НЕТ")]
    public string YesNo { get; set; }
    public virtual ICollection<_0YesNo> _0YesNos { get; set; }
}
```

Приведенная модель и логика предметной области позволит в дальнейшем создать базу данных с помощью инфраструктуры Entity Framework. Подобное действие будет выполняться, автоматически не требуя понимания и технологии работы Entity Framework. Для базы данных будет применяться SQL сервер. Доступ к базе данных будет осуществляться с помощью Entity Framework (EF). Эти и другие действия по созданию сайта рассмотрим в следующих публикациях.

Список использованной литературы

Зиборов В.В. Visual C# 2012 на примерах. СПб.: БХВ-Петербург, 2013. 480 с. (twirpx.com>Все для студента>Code).

Мак-Дональд М., Фримен А., Шнушта М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов. 4-е изд.: Пер. с англ. М.: ООО «И.Д. Вильямс», 2011. 1424 с. (alltorrents.net>viewtopic.php?t=19120).

Макки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов.: Пер. с англ. М.: ООО «И.Д. Вильямс», 2010. 416 с. (mexalib.com>view/334).

Столбовский Д.Н. Основы разработки Web-приложений на ASP.NET: Учебное пособие. М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2011. 304 с. (Основы информационных технологий). (librabook.com.ua>121/Osnovy...na-ASP-NET-8951.html).

Фримен А. MVC 5 с примерами на C# для профессионалов. 5е изд.: Пер. с англ. М.: ООО «И.Д. Вильямс», 2015а. 736 с.

Фримен А. jQuery для профессионалов. 5-е изд.: Пер. с англ. М.: ООО «И.Д. Вильямс», 2015б. 1040 с. (williamspublishing.com>cgi-bin/list.cgi?id=1070862).

Фримен А., Сандерсон С. ASP.NET MVC 3 Framework с примерами на C# для профессионалов. 3-е изд.: Пер. с англ. М.: ООО «И.Д. Вильямс», 2012. 672 с. (dom-eknig.ru>...aspnet-mvc-3-framework...primerami-na...).

Часовских В.П., Стаин Д.А. Структура, содержание и среда разработки веб-сайта вуза // Эко – Потенциал: журнал мультидисциплинарных научных публикаций. 2013. № 3-4. С. 160-173 (<http://elar.usfeu.ru/handle/123456789/2739>).

Часовских В.П., Стаин Д.А. Представление и сравнительный анализ модели образовательного процесса университета в форме реляционных таблиц SQL и их мультипликативное отображение // Техника и технология: новые перспективы развития, научный журнал «Естественные и технические науки». Москва. 2014а. С. 101-106.

Часовских В.П., Стаин Д.А. Модель образовательного процесса и сайт вуза 2.0 // Эко–Потенциал: журнал мультидисциплинарных научных публикаций. Екатеринбург. 2014б. № 2(6). С. 113-119 (<http://elar.usfeu.ru/bitstream/123456789/3362/1/Chasovskikh.pdf>).

Эспозито Д. Программирование на основе Microsoft ASP.NET MVC. 2-е изд. / Пер. с англ. М. : Издательство «Русская редакция» ; СПб.: БХВ-Петербург, 2012. 464 с. («Русская редакция») ([mdk-arbat.ru>bookcard?book_id=755050](http://mdk-arbat.ru/bookcard?book_id=755050)).

Эспозито Д. Программирование с использованием Microsoft ASP.NET 4. СПб.: Питер, 2013. 880 с. ([bookvoed.ru>book?id=613853](http://bookvoed.ru/book?id=613853)).

Рецензент статьи: кандидат технических наук,
профессор кафедры менеджмента и ВЭДП
Уральского государственного лесотехнического университета М.П. Воронов.